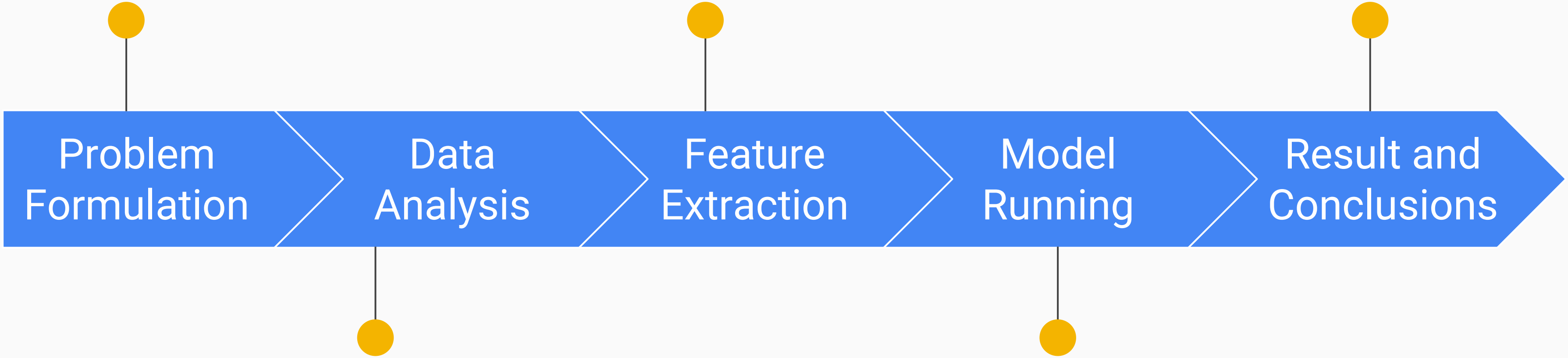
Winner Prediction in IPL Match

Gaurav Anand
Shivam Shakti
Sunil Kumar
Vishal Kumar

Dataset

- From Kaggle.
- Ball by ball details for all IPL matches between 2008 - 2017.
- Matches.csv file has information about the match venue, result, toss, umpire etc...
- Deliveries.csv file has ball by ball data of all the matches.
- 636 matches.

Phases



Problem Formulation

We formulate our Problem in a formal way.

Data Analysis

Analysis of data is done to understand the important features.

Feature Extraction

Features are extracted and new features are formed.

Model Running

Run different models and compare the result.

Result and Conclusions

Results of previous phases are compared and concluded.

Problem Formulation

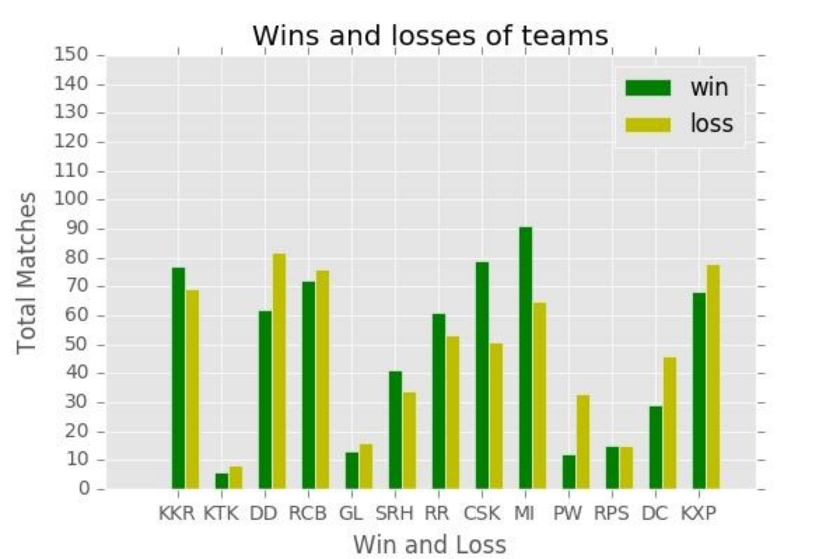
Given

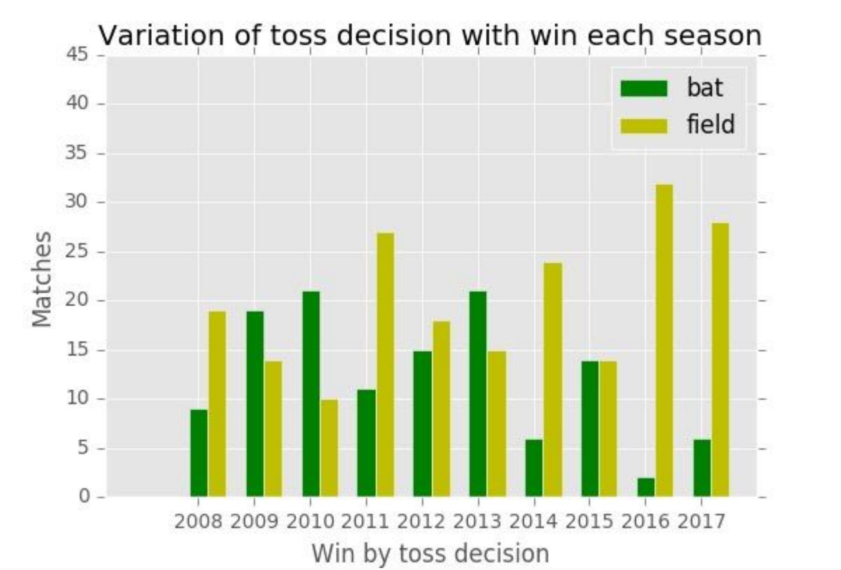
m	*Match*
T_i	I^{th} *team playing* m, $i \in \{1,2\}$
$O(m)$	*Outcome of toss of* m
$L(m)$	*Venue of* m
$H(T_1, T_2)$	*History of of the matches played between* T_1 *and* T_2
$P(T,m)$	*Set of all players in team* T *playing* m
$C(p,m)$	*Career statistics of* p *playing* m

Predict

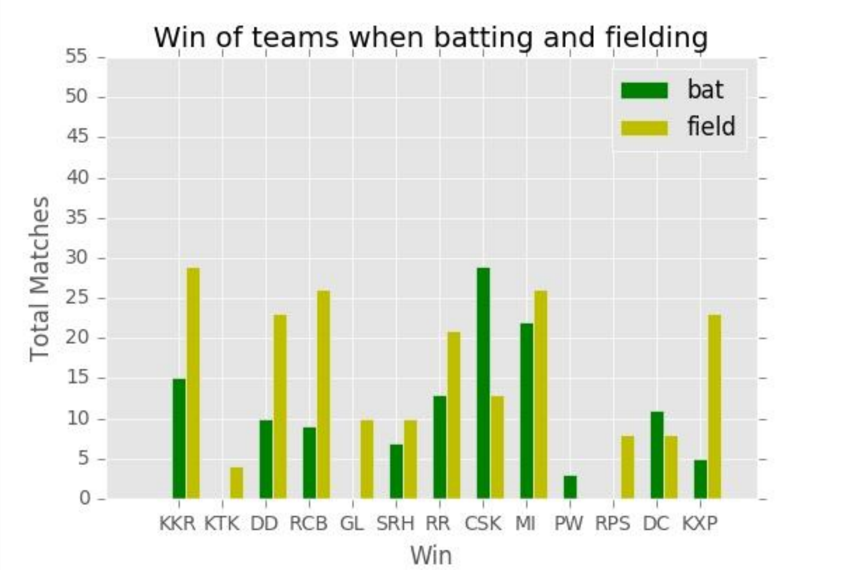
$W(m)$	*Winner of* m

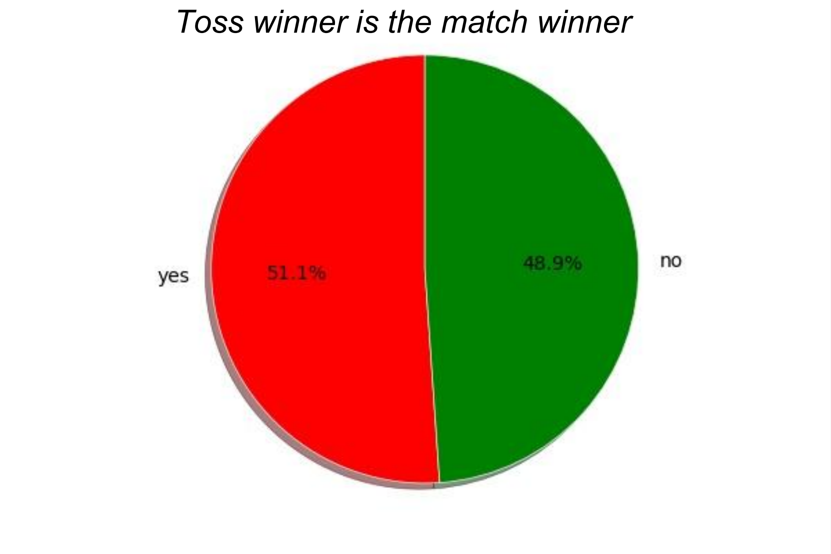
Data Analysis



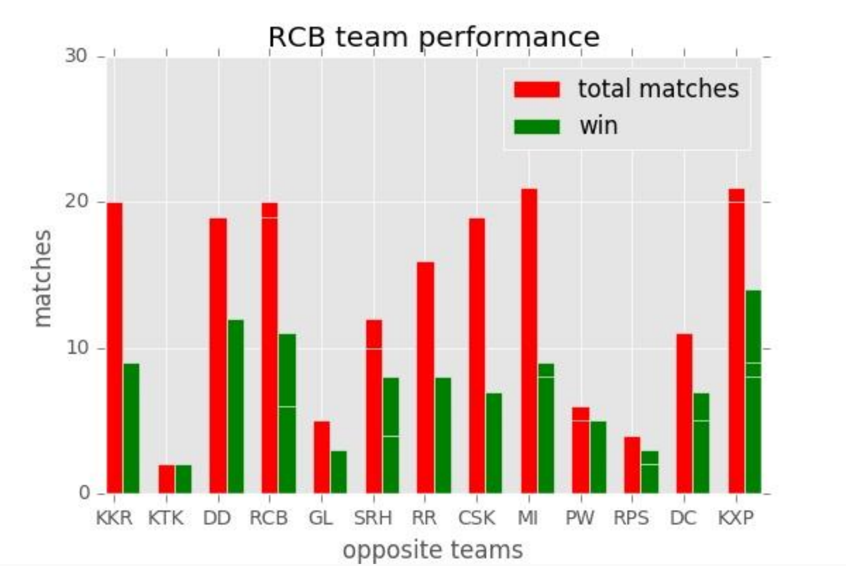


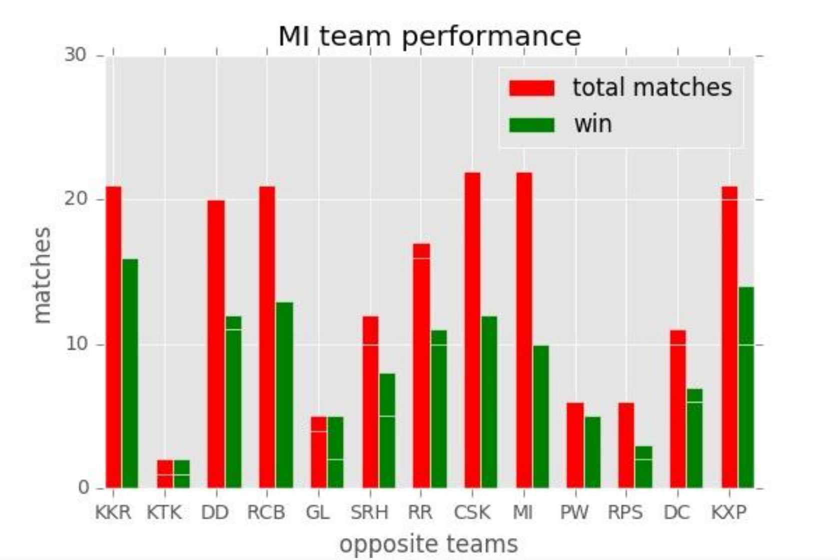
Data Analysis ...





Data Analysis ...





Feature Extraction

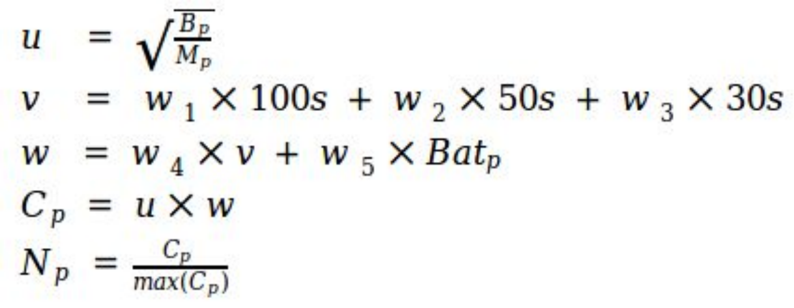
01	**Teams**	Team 1 and Team 2
02	**Venue**	
03	**Toss Outcome**	
04	**Previous History**	Outcome of previous matches between both the teams
05	**Umpire**	We can drop this feature but we were curious to observe

New Feature Development

Model a Batsman, p

M_p	*Total matches played by p*
B_p	*Total innings p had batted in*
R_p	*Total runs scored by p*
Bat_p	*Batting average of p*
100s	*Total centuries by p*
50s, 30s	*Total half centuries and 30s by p*

Method



$$
\begin{aligned}
u &= \sqrt{\frac{B_p}{M_p}} \\
v &= w_1 \times 100s + w_2 \times 50s + w_3 \times 30s \\
w &= w_4 \times v + w_5 \times Bat_p \\
C_p &= u \times w \\
N_p &= \frac{C_p}{max(C_p)}
\end{aligned}
$$

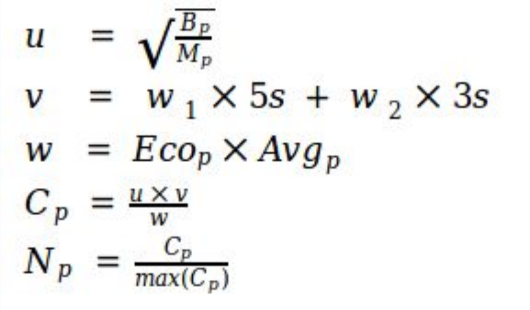
C_p	*Career score of p*
N_p	*Normalised Career score of p*

New Feature Development ...

Model a Bowler, p

M_p	*Total matches played by p*
B_p	*Total innings p had bowled in*
W_p	*Total wickets taken by p*
Eco_p	*Economy of p*
Avg_p	*Bowl average of p*
5s, 3s	*Total 5 and 3 wicket hauls by p*

Method



$$\begin{aligned} u &= \sqrt{\frac{B_p}{M_p}} \\ v &= w_1 \times 5s + w_2 \times 3s \\ w &= Eco_p \times Avg_p \\ C_p &= \frac{u \times v}{w} \\ N_p &= \frac{C_p}{max(C_p)} \end{aligned}$$

C_p	*Career score of p*
N_p	*Normalised Career score of p*

New Feature Development ...

$$Bat_A = \sum_{p \,\epsilon\, P(A,m)} N_{p_{bat}}$$

$$Bowl_A = \sum_{p \,\epsilon\, P(A,m)} N_{p_{bowl}}$$

$$Bat_B = \sum_{p \,\epsilon\, P(B,m)} N_{p_{bat}}$$

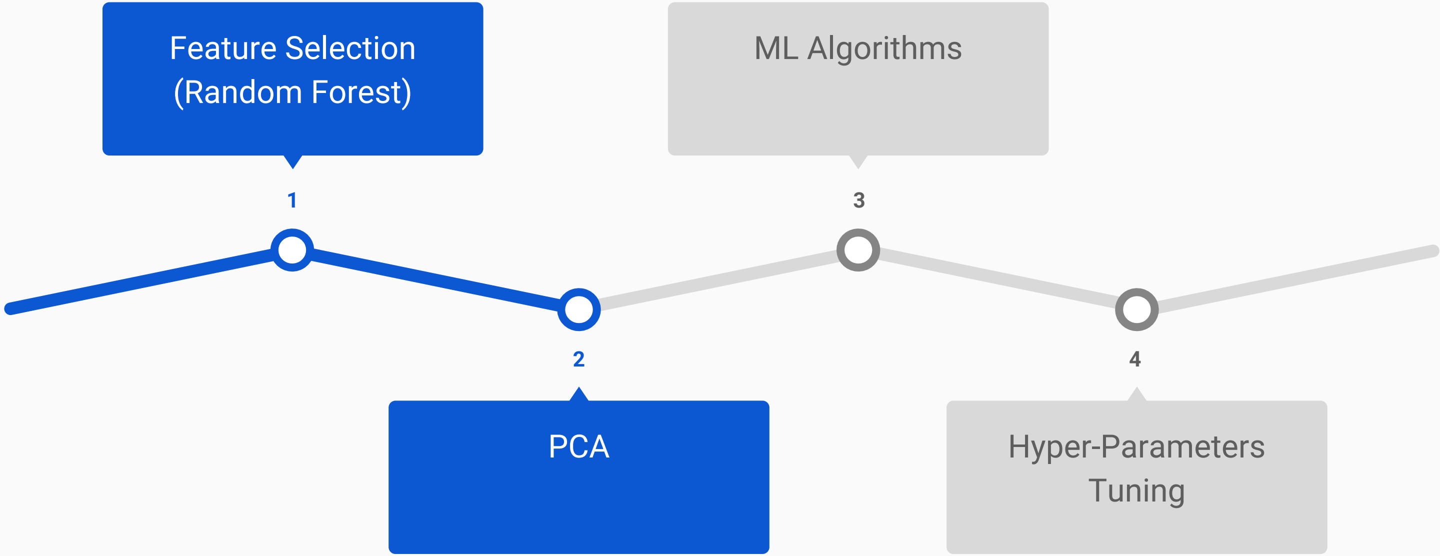
$$Bowl_B = \sum_{p \,\epsilon\, P(B,m)} N_{p_{bowl}}$$

Strength of team A wrt B

$$SAB = \frac{Bat_A}{Bowl_B} - \frac{Bat_B}{Bowl_A}$$

➢ SAB is our new feature

Training and Testing



1 Feature Selection (Random Forest)

2 PCA

3 ML Algorithms

4 Hyper-Parameters Tuning

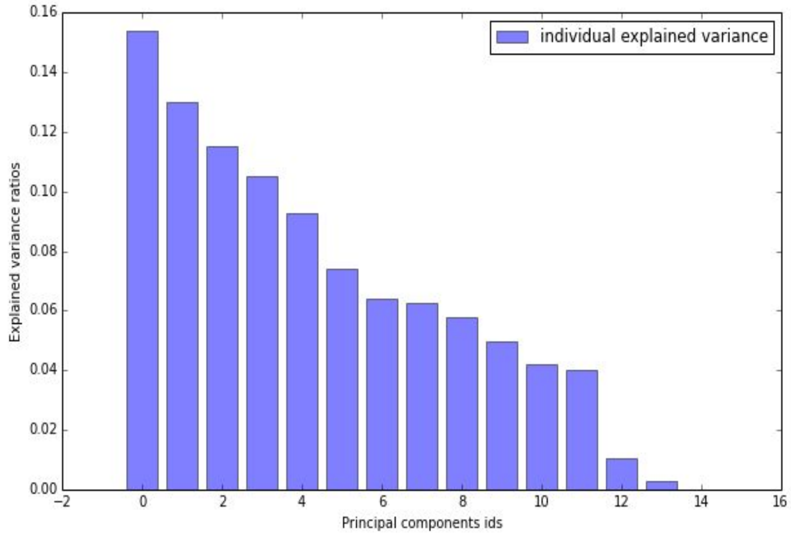
Training and Testing ...

Feature Selection :- We used Random Forest Classifier

Around 5 features were removed.

PCA :

Features	*Variance(%)*
First 6	*~62*
7-12	*~33*
Rest	*< 5%*



Model Comparisons

54%

Random Forest

Folds = 10

- bootstrap=True
- min samples leaf=5
- n estimators=2000

55%

Decision Tree

Folds = 10

- Criterion': entropy
- Max_depth': 5
- max_features':None

58%

Logistic Regression

Folds = 10

- C' : 0.01
- Solver' : sag

56%

K Nearest Neighbour

Folds = 10

- leaf_size': 1
- 'metric': ' cityblock'
- N_neighbors': :29

58%

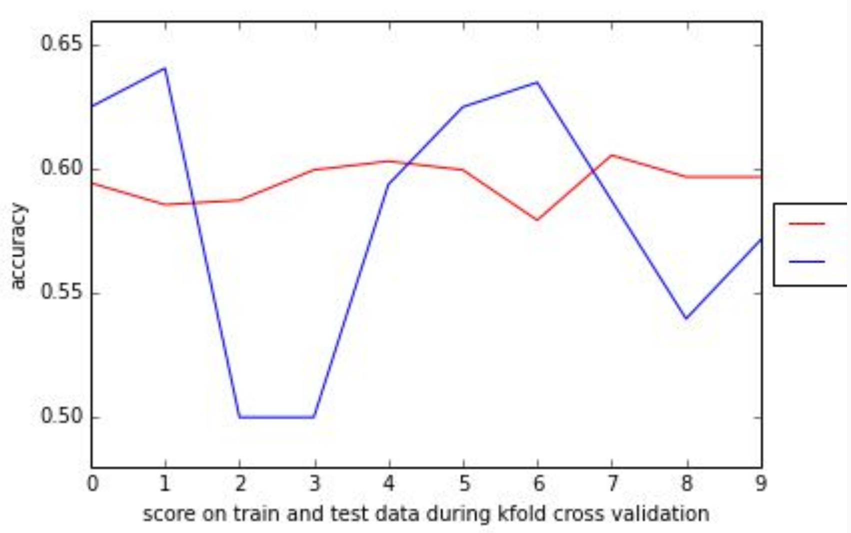
MLP

Folds = 10

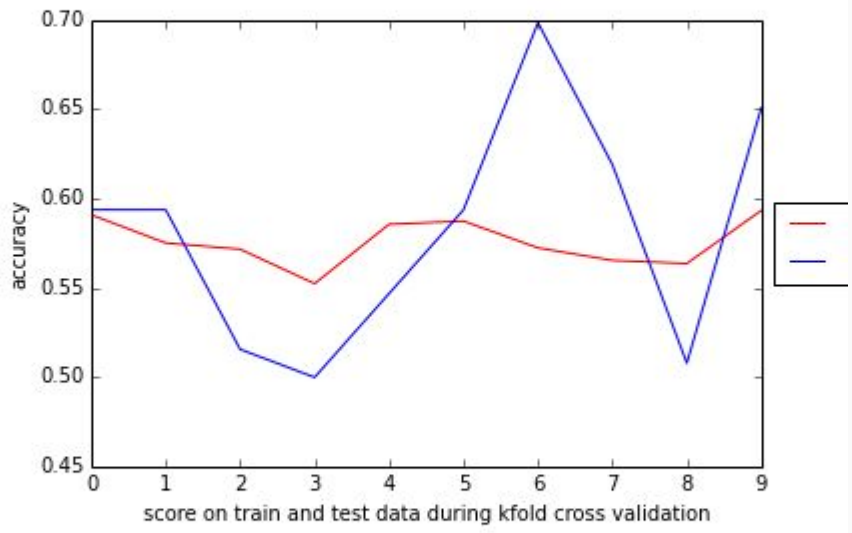
- Solver:'sgd',
- Alpha:1e3,
- hidden layer sizes:(9)

Classifier Accuracy and Cross Validation

Logistic Regression



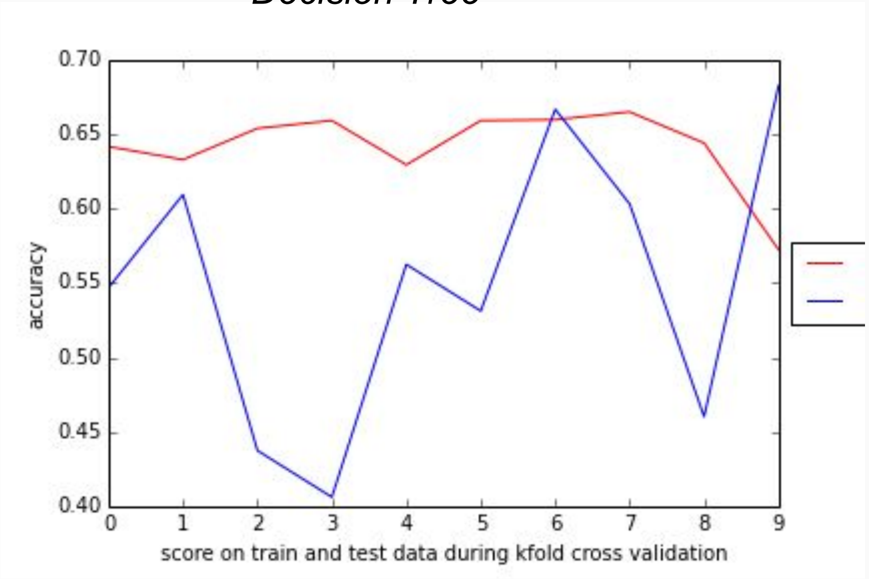
MLP



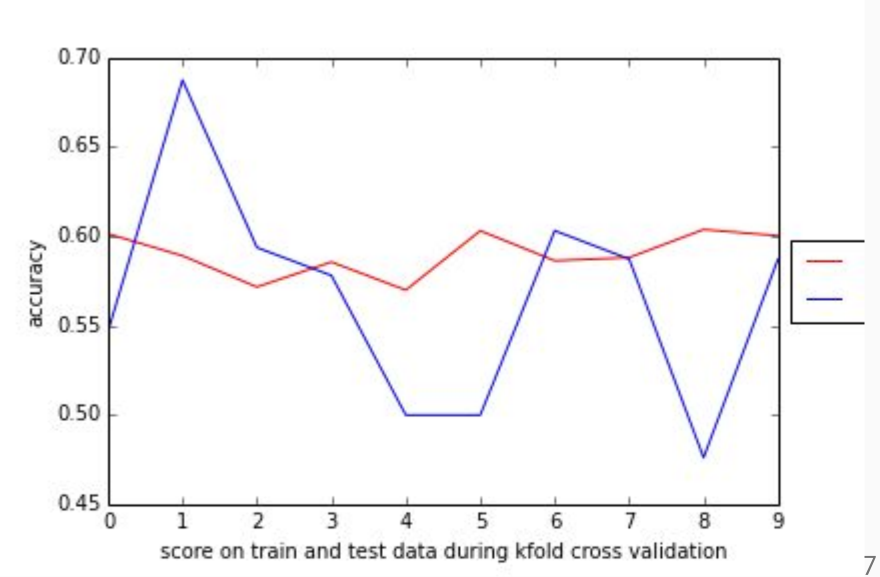
Note : Red line is testing on training data and blue line is testing on test data

Classifier Accuracy and Cross Validation

Decision Tree



SVM



Conclusion

1. Problem of determining the winner of cricket match.
2. Key Features: Players statistics and team history.
3. Prediction Score ~ 55%
4. Best score on Kaggle = 49%, so our model beats that.

Thankyou !